UdDA – Vzorové riešenia príkladov z písomky

Ivan Kováč

$\boxed{\mathbf{0}}$ Potrebujeme nájsť algoritmus, pomocou ktorého oba aktívne uzly (označme ich A a B) doručia správu (svoj identifikátor) do toho istého uzla (označme ho C), aby tento uzol mohol poslať identifikátor A do uzla B a naopak. Navyše sme obmedzení komunikačnou zložitosťou, ktorá má byť $O(\sqrt{n}) = O(2^{\frac{d}{2}})$.

Ľahko nahliadneme, že ak do uzla C dokážeme dostať identifikátory pomocou $O(2^{\frac{d}{2}})$ krokov, tak aj z uzla C vieme uzlom A a B poslať identifikátor druhého aktívneho uzla pomocou $O(2^{\frac{d}{2}})$ krokov – stačí si v správe pamatať po ktorých dimenziách sme sa pohybovali a rovnako sa vracať naspäť. Problémom ostáva ako dostať identifikátory z A a B do C na $O(2^{\frac{d}{2}})$ krokov. Najprv skúsme zistiť, ktorý vrchol je vhodný kandidát na C. Vrcholy hyperkocky môžeme označiť d-ticou bitov, nech platí $A = a_1a_2 \dots a_d$ a $B = b_1b_2 \dots b_d$. Pre uzol C potom bude platiť $C = a_1a_2 \dots a_{\lfloor \frac{d}{2} \rfloor} b_{\lfloor \frac{d}{2} \rfloor + 1} \dots b_d$. Skúsme teraz poslať identifikátor vrchola A do C pomocou $O(\sqrt{n})$ krokov. Vrchol A musí vygenerovať celú cestu dĺžky $2^{\lfloor \frac{d}{2} \rfloor}$. Môže to urobiť napríklad takto

$$\begin{aligned}
P_1 &= 1, \\
P_2 = P_1, 2, P_1 &= 1, 2, 1, \\
P_3 = P_2, 3, P_2 &= 1, 2, 1, 3, 1, 2, 1, \\
&\vdots \\
P_{\lfloor \frac{d}{2} \rfloor} &= P_{\lfloor \frac{d}{2} \rfloor - 1}, \left\lfloor \frac{d}{2} \right\rfloor, P_{\lfloor \frac{d}{2} \rfloor - 1}
\end{aligned}$$

Ak vrchol A pošle po takejto ceste správu so svojim identifikátorom, táto dôjde do každého vrchola ktorý má s A spoločnú druhú polovicu bitov (súradníc). Podobne môžeme vygenerovať cestu $R^d_{\lfloor \frac{d}{2} \rfloor + 1}$ ktorá bude chodiť po nohách od $\lfloor \frac{d}{2} \rfloor + 1$ po d. Zjavne dĺžky oboch ciest sú $O(\sqrt{n})$ a teda pomocou $O(\sqrt{n})$ správ dostaneme identifikátory A aj B do C a späť. (Aby vrchol ktorému príde správa `<id, cesta>` vedel kam má pokračovať, musí mať v správe označené kde sa nachádza, napríklad tak, že správa bude obsahovať aj kompletnú cestu aj ostávajúcu časť cesty.) Jediným problémom ostáva zistiť, ktorý z aktívnych vrcholov je vrchol A a ktorý B. Keďže to na začiatku nevedia, budú sa oba aktívne vrcholy správať aj ako vrchol A aj ako vrchol B, teda pošlú svoj identifikátor aj po ceste P aj po ceste R. Tým dostaneme dva vrcholy C, čo nám však neprekáža. Komunikačná zložitosť zostáva asymptoticky rovnaká.

$\boxed{\mathbf{1}}$ neviem

$\boxed{\mathbf{2}}$ Pozrime sa na k. kolo. V tomto kole sa narodí aspoň $pn2^n > 2^{n+1}$ nových paketov. Vezmime si rez nejakou dimenziou. (Pevne zvolenou dimenziou hyperkocky z ktorej graf vznikol.) Tento rez rozdelí vrcholy na dve rovnako veľké skupiny. Označme počet paketov ktoré žili aj v $k-1$. kole a chcú prejsť z jednej skupiny do druhej a_{k-1}. Z nových paketov chce približne polovica prejsť z jednej skupiny do druhej. Z toho dostávame, že viac ako $a_{k-1} + 2^n$ paketov chce prejsť v k. kole z jednej skupiny do druhej. Počet hrán medzi týmito dvoma skupinami je z definície siete 2^{n-1} teda ak uvažujeme full duplex v k. kole môže dimenziu zmeniť najviac 2^n paketov. Z toho vyplýva, že $a_k > a_{k-1} + 2^n - 2^n$, teda počet správ, ktoré nestihnú zmeniť dimenziu rastie. Z toho ale vyplýva, že týchto správ bude pre k idúce do nekonečna nekonečne veľa, teda aj veľkosť buffrov na zapamätanie týchto paketov musí ísť do nekonečna.

$\boxed{\mathbf{3}}$ Označme partície bipartitného grafu $K_{n,n}$ písmenami A a B. Využijeme algoritmus voľby šéfa na úplnom grafe popísanom na prednáške, ktorého komunikačná zložitosť je $O(n \log n)$. Ním zvolíme šéfa v partícii A tak, že každý vrchol v z A si zvolí jednu svoju nohu za komunikačnú a ak chce v rámci algoritmu na úplnom grafe poslať správu po svojej i-tej nohe, pošle po svojej komunikačnej nohe pôvodnú správu a číslo i, vrchol z B ktorý takúto správu dostane, pošle pôvodnú správu po svojej i-tej nohe. (S tým, že pri očíslovaní svojich nôh vynechá tú spojenú s v.) Rovnako bude fungovať aj algoritmus v rámci partície B. Keď obe simulácie voľby šéfa na úplnom grafe skončia, bude práve jeden šéf v partícii A a práve jeden šéf v partícii B. Títo pomocou $O(n)$ správ zistia kto je naozaj šéf tak, že pošlú svoje identifikátory po každej svojej nohe (napríklad v tvare `<myID, boss?>`) a teda aj sebe navzájom, potom ten z nich, ktorý ma väčší identifikátor pošle každému správu `<myID, som boss>` a porazený pošle každému okrem šéfa správu `<bossID, je boss>`. Komunikačná zložitosť ostáva $O(n \log n)$. Teraz ukážeme, že je asymptoticky najlepšia. Ak by totiž existoval lepší algoritmus, vedeli by sme ho upraviť na lepší algoritmus pre voľbu šéfa na úplnom grafe, čo z prednášky nevieme. Každý vrchol úplneho grafu by mohol simulovať dva vrcholy, jeden z partície A, druhý z partície B, pričom identifikátor v partícii A by bol jeho identifikátor $myID$ a v partícii B by to bol $-myID$. Tým by sme sa vyhli kolízii identifikátorov a zvolili by sme v lepšom čase ako $O(n \log n)$ šéfa na úplnom grafe.